ESKE 3 C.53 Mycro-1

# MYCRO-1

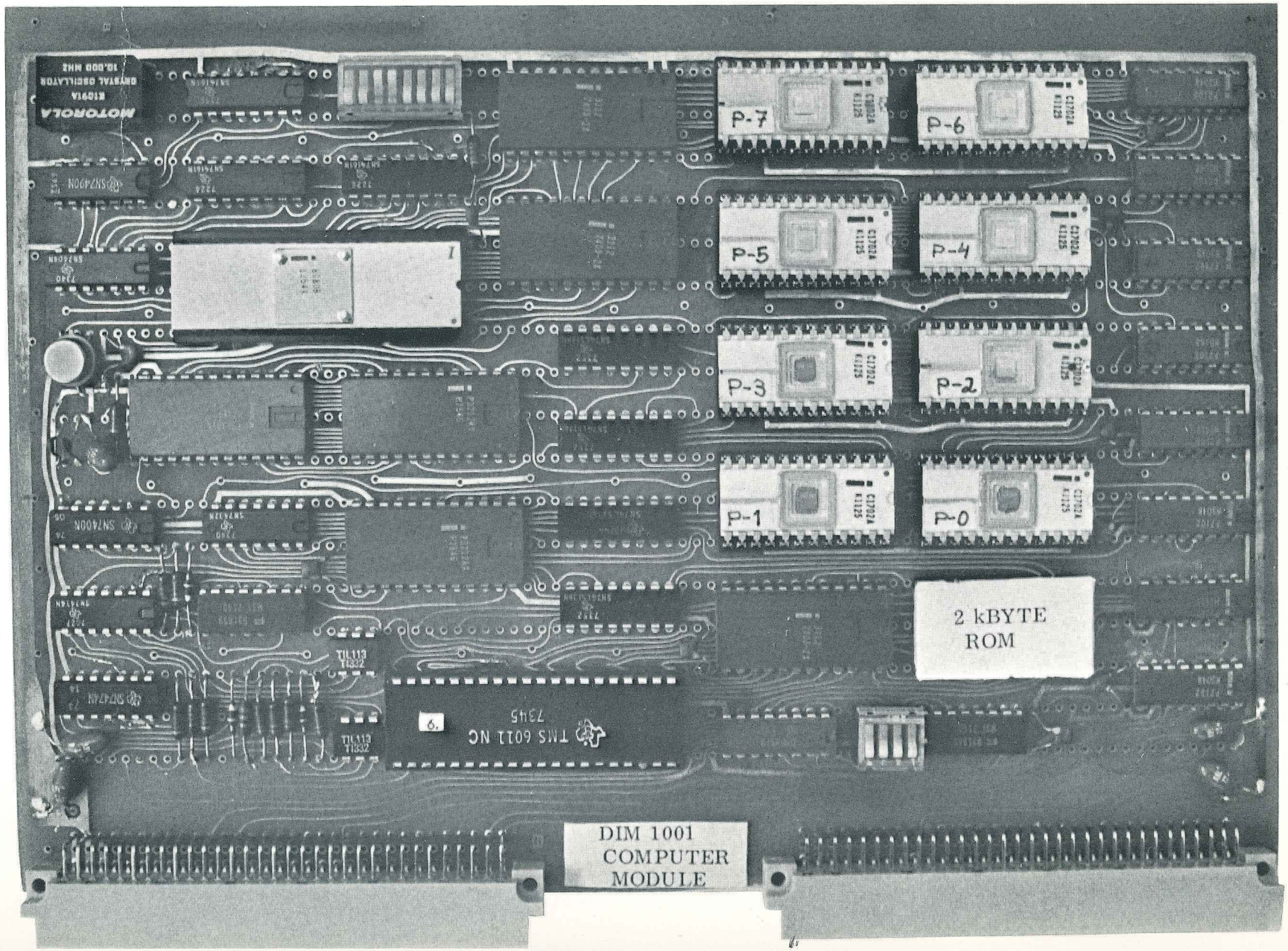
# MICROCOMPUTER

A/S MYCRON

P.O. Box 6199, Etterstad, Oslo 6

Norway

Universitetet i Oslo
Informatikkbiblioteket
Postboks 1126 Blindern
0316 OSLO


P-7
P-6
P-5
P-4
P-3
P-2
P-1
P-0
2 kBYTE
ROM
DIM 1001
COMPUTER
MODULE

07 ni 05052

## MYCRO-1 MICROCOMPUTER:

The MYCRO-1 is a byte-oriented computer system designed for general purpose and dedicated system applications. The functional, mechanical and electrical design is the basis for a series of machine configurations built with standard function modules.

The design is modular to suit a vast number of applications, and provides a simple and inexpensive solution to the given problem without loss of efficiency and expansion capability.

The basic modules in the system MYCRO-1 are:

| | |
|---|---|
| DIM-1001 | Computer Module |
| DIM-1010 | 4kByte RAM Module |
| DIM-1012 | 4kByte PROM Module |
| DIM-1013 | 16kByte RAM Module |
| DIM-1021 | Quad Peripheral Input/Output Module |
| DIM-1030 | Disc Controller (4 x 256kByte) |
| DIM-1090 | Chassis with Motherboard |
| DIM-1091 | Power Supply |

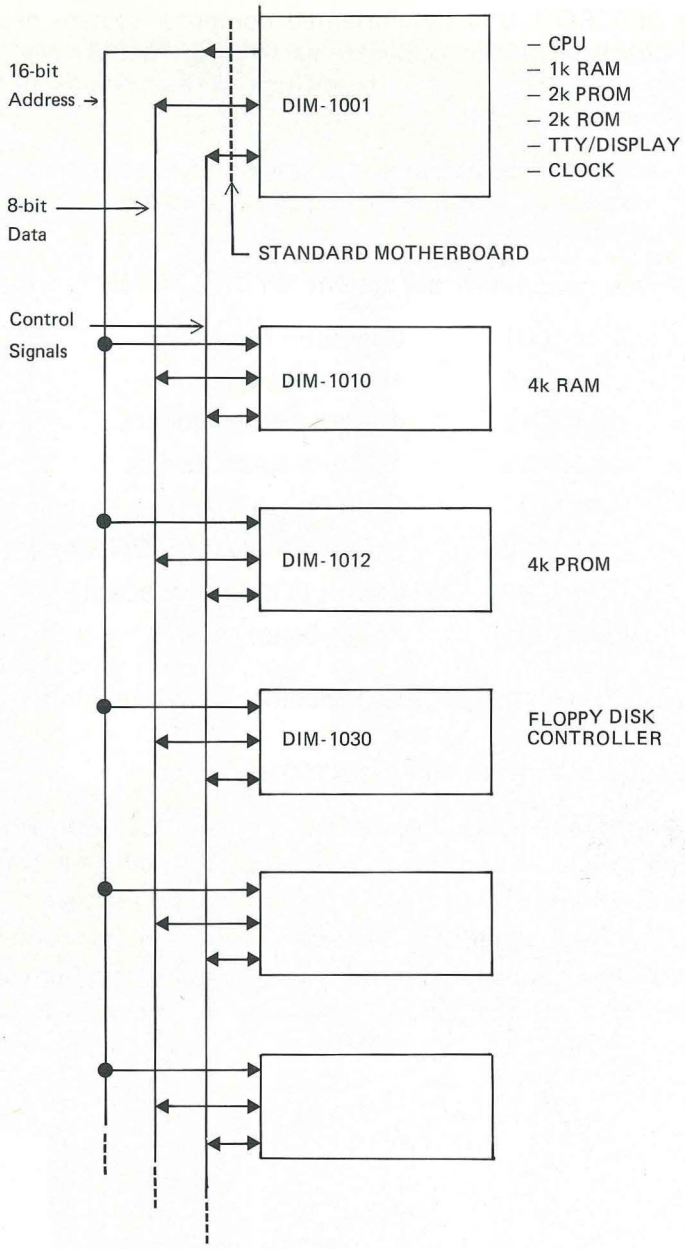
Process Control Input/Output Modules are covered in a separate publication.

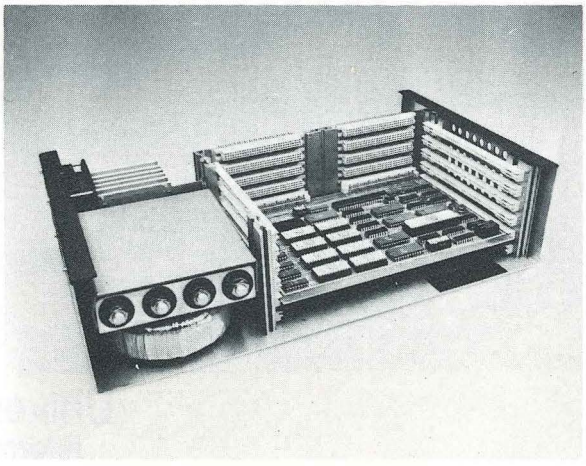
## TYPICAL MYCRO-1 APPLICATIONS

- Dedicated Business Calculation
- Point-of-Sale Terminals
- Text Editing Systems
- Distributed Computing Systems
- Industrial Sequence Control
- Medical Instrumentation
- Classification and Inventory Control
- Intelligent Terminals
- Instruction
- Process Monitoring
- Process Control
- Industrial Robot Control

Universitetet i Oslo
Informatikkbiblioteket
Postboks 1128 Blindern
0316 OSLO

MYCRO-1


16-bit
Address
8-bit
Data
Control
Signals
DIM-1001
– CPU
– 1k RAM
– 2k PROM
– 2k ROM
– TTY/DISPLAY
– CLOCK
STANDARD MOTHERBOARD
DIM-1010
4k RAM
DIM-1012
4k PROM
DIM-1030
FLOPPY DISK
CONTROLLER



## DIS 1001 PROCOM

The MYCRO-1 is intended to be an element in end user systems. However, for program development and test, a means for debugging on the "register level" is a must. For this purpose DI has developed a programmer communication system for use via teletype or compatible display. PROCOM is a program stored in PROM in the lowest part of memory space.

Highlights of the PROCOM program are:

- PROCOM is started by either: Master Clear interrupt (automatic at power up), — Teletype interrupt, — or by Call for PROCOM from the user's program.
- Jump out of PROCOM is caused by a GO command on the following assumptions that 1) Jump to PROCOM was caused by teletype interrupt or by a call from user's program or by a break point set by PROCOM. The execution of user's program will proceed from where it was interrupted. 2) A GO command with address has been given.
- All CPU-registers and condition flags as CARRY, ZERO, SIGN, PARITY and CARRY for "desimal adjust" can be read out and a new value can be put in. A, B, C, D, E, H and L as single byte and HL called M, Stack pointer and Program counter (the return address to user program) as double byte. All registers can be dumped out together.
- All memory locations can be read out or a new value can be put in. Furthermore a specified block can be dumped out.
- Programs can be read into memory from paper tape and memory content can be dumped out on paper tape on the teletype if teletype is used.
  A hexadecimal format is used.
- By GO commands with start address specification and break points, one or two, program execution can be started and interrupted at any place wanted.
- An assign command permits the user to assign physical devices to one of four logical devices, named CONSOLE an input/output device, READER an input-only device, PUNCH an output-only device and LIST an output-only device.
- A command enables the user to fill a block of memory with CONSTANT.
- Data can, via teletype or display with keyboard, be read in or written out, hexadecimally or each byte as an ASCII-character.
- PROCOM requires DIM-1001 Computer Module, DIP-1023 Teletype or DIP-1022 Display.

## DIS-1002 MYCRA ONE-PASS ASSEMBLER

MYCRA is a one-pass assembler running on MYCRO-1 which reads a symbolic source program, translates it, and generates an absolute object program for the DIM-1001 computer module. The MYCRO-1 assembler is designed to be a debugging aid for interactive use and allows error recovery during and after actual assembly.

Highlights of the MYCRA program are:

- Performs one-pass assembly in two modes:
  1. On-line with input from console/teletype
  2. On-line with input from tape reader (batch mode)
- Listing is optional when operating in batch mode thus giving error-only listing at assembly time. (High-speed processing.)
- The generated object code is either
  - stored directly into memory for execution and debugging, or
  - punched out on paper tape in a hexadecimal format, which can be loaded by the monitor PROCOM, or
  - both stored into memory and punched out on paper tape simultaneously, or
  - no code output, only syntax checking
- Each source statement may be written in free format.
- Symbolic addressing is permitted and both defined and undefined symbols are allowed in constants. Expressions may consist of symbols and octal, decimal and hexadecimal numbers (and the binary operators + and −).
- The symbol tables may be placed anywhere in RAM memory and with variable size.
- A set of pseudo instructions are available in order to direct the assembly process, like ORG (define origin), EQU (define symbol value), DB (define byte), DW (define word) and DS (define storage).
- A set of assembly commands to control the information flow of the source program, listing and object code, and to help with the debugging process. The commands are as follows:

| | |
|---|---|
| .I | Initialization of assembler and symbol tables |
| .A R = PTR<br>.A L = NONE<br>.A P = PTP | Assignment of logical reader (source), list and punch (object) to available physical devices.<br>R, L, P = NONE, TTY, PTR, LP, CR, MEMO, etc. |
| .R | Start reading source program from assigned fast reader |
| . | Break the assembly processing (in batch mode) |
| .C | Continue the assembly processing |
| .S | List the defined symbols with corresponding values |
| .U | List the undefined symbols and reference addresses |
| .P | Call to the monitor system PROCOM |
| .G LABEL | Start execution of program at address LABEL |

- The assembler resides in approx. 3kByte of memory on the DIM-1012-1, DIM-1010-2 or other memory cards.
- The source program format is fully compatible with the intellex-8 Mod 80 assembler with the exception of macros and expression complexity.

### JUMP

| | | |
|---|---|---|
| C3 | JMP | Adr |
| C2 | JNZ | |
| CA | JZ | |
| D2 | JNC | |
| DA | JC | |
| E2 | JPO | |
| EA | JPE | |
| F2 | JP | |
| FA | JM | |
| E9 | PCHL | |

### CALL

| | | |
|---|---|---|
| CD | CALL | Adr |
| C4 | CNZ | |
| CC | CZ | |
| D4 | CNC | |
| DC | CC | |
| E4 | CPO | |
| EC | CPE | |
| F4 | CP | |
| FC | CM | |

### RETURN

| | |
|---|---|
| C9 | RET |
| C0 | RNZ |
| C8 | RZ |
| D0 | RNC |
| D8 | RC |
| E0 | RPO |
| E8 | RPE |
| F0 | RP |
| F8 | RM |

### RESTART

| | | |
|---|---|---|
| C7 | RST | 0 |
| CF | RST | 1 |
| D7 | RST | 2 |
| DF | RST | 3 |
| E7 | RST | 4 |
| EF | RST | 5 |
| F7 | RST | 6 |
| FF | RST | 7 |

### ROTATE†

| | |
|---|---|
| 07 | RLC |
| 0F | RRC |
| 17 | RAL |
| 1F | RAR |

### CONTROL

| | |
|---|---|
| 00 | NOP |
| 76 | HLT |
| F3 | DI |
| FB | EI |

### MOVE (cont)

| | | |
|---|---|---|
| 58 | MOV | E,B |
| 59 | MOV | E,C |
| 5A | MOV | E,D |
| 5B | MOV | E,E |
| 5C | MOV | E,H |
| 5D | MOV | E,L |
| 5E | MOV | E,M |
| 5F | MOV | E,A |
| 60 | MOV | H,B |
| 61 | MOV | H,C |
| 62 | MOV | H,D |
| 63 | MOV | H,E |
| 64 | MOV | H,H |
| 65 | MOV | H,L |
| 66 | MOV | H,M |
| 67 | MOV | H,A |
| 68 | MOV | L,B |
| 69 | MOV | L,C |
| 6A | MOV | L,D |
| 6B | MOV | L,E |
| 6C | MOV | L,H |
| 6D | MOV | L,L |
| 6E | MOV | L,M |
| 6F | MOV | L,A |
| 70 | MOV | M,B |
| 71 | MOV | M,C |
| 72 | MOV | M,D |
| 73 | MOV | M,E |
| 74 | MOV | M,H |
| 75 | MOV | M,L |
| | --- | --- |
| 77 | MOV | M,A |
| 78 | MOV | A,B |
| 79 | MOV | A,C |
| 7A | MOV | A,D |
| 7B | MOV | A,E |
| 7C | MOV | A,H |
| 7D | MOV | A,L |
| 7E | MOV | A,M |
| 7F | MOV | A,A |

### ACCUMULATOR*

| | | |
|---|---|---|
| 80 | ADD | B |
| 81 | ADD | C |
| 82 | ADD | D |
| 83 | ADD | E |
| 84 | ADD | H |
| 85 | ADD | L |
| 86 | ADD | M |
| 87 | ADD | A |
| 88 | ADC | B |
| 89 | ADC | C |
| 8A | ADC | D |
| 8B | ADC | E |
| 8C | ADC | H |
| 8D | ADC | L |
| 8E | ADC | M |
| 8F | ADC | A |
| 90 | SUB | B |
| 91 | SUB | C |
| 92 | SUB | D |
| 93 | SUB | E |
| 94 | SUB | H |
| 95 | SUB | L |
| 96 | SUB | M |
| 97 | SUB | A |
| 98 | SBB | B |
| 99 | SBB | C |
| 9A | SBB | D |
| 9B | SBB | E |
| 9C | SBB | H |
| 9D | SBB | L |
| 9E | SBB | M |
| 9F | SBB | A |
| A0 | ANA | B |
| A1 | ANA | C |
| A2 | ANA | D |
| A3 | ANA | E |
| A4 | ANA | H |
| A5 | ANA | L |
| A6 | ANA | M |
| A7 | ANA | A |
| A8 | XRA | B |
| A9 | XRA | C |
| AA | XRA | D |
| AB | XRA | E |
| AC | XRA | H |
| AD | XRA | L |
| AE | XRA | M |
| AF | XRA | A |
| B0 | ORA | B |
| B1 | ORA | C |
| B2 | ORA | D |
| B3 | ORA | E |
| B4 | ORA | H |
| B5 | ORA | L |
| B6 | ORA | M |
| B7 | ORA | A |
| B8 | CMP | B |
| B9 | CMP | C |
| BA | CMP | D |
| BB | CMP | E |
| BC | CMP | H |
| BD | CMP | L |
| BE | CMP | M |
| BF | CMP | A |

### MOVE IMMEDIATE

| | | |
|---|---|---|
| 06 | MVI | B, D8 |
| 0E | MVI | C, |
| 16 | MVI | D, |
| 1E | MVI | E, |
| 26 | MVI | H, |
| 2E | MVI | L, |
| 36 | MVI | M, |
| 3E | MVI | A, |

### Acc IMMEDIATE*

| | | |
|---|---|---|
| C6 | ADI | D8 |
| CE | ACI | |
| D6 | SUI | |
| DE | SBI | |
| E6 | ANI | |
| EE | XRI | |
| F6 | ORI | |
| FE | CPI | |

### LOAD IMMEDIATE

| | | |
|---|---|---|
| 01 | LXI | B, D16 |
| 11 | LXI | D, |
| 21 | LXI | H, |
| 31 | LXI | SP, |

### STACK OPS

| | | |
|---|---|---|
| C5 | PUSH | B |
| D5 | PUSH | D |
| E5 | PUSH | H |
| F5 | PUSH | PSW |
| C1 | POP | B |
| D1 | POP | D |
| E1 | POP | H |
| F1 | POP | PSW* |
| E3 | XTHL | |
| F9 | SPHL | |

### DOUBLE ADD†

| | | |
|---|---|---|
| 09 | DAD | B |
| 19 | DAD | D |
| 29 | DAD | H |
| 39 | DAD | SP |

### SPECIALS

| | |
|---|---|
| EB | XCHG |
| 27 | DAA* |
| 2F | CMA |
| 37 | STC† |
| 3F | CMC† |

### LOAD/STORE

| | | |
|---|---|---|
| 0A | LDAX | B |
| 1A | LDAX | D |
| 2A | LHLD | Adr |
| 3A | LDA | Adr |
| 02 | STAX | B |
| 12 | STAX | D |
| 22 | SHLD | Adr |
| 32 | STA | Adr |

### INPUT/OUTPUT

| | | |
|---|---|---|
| D3 | OUT | D8 |
| DB | IN | D8 |

### INCREMENT**

| | | |
|---|---|---|
| 04 | INR | B |
| 0C | INR | C |
| 14 | INR | D |
| 1C | INR | E |
| 24 | INR | H |
| 2C | INR | L |
| 34 | INR | M |
| 3C | INR | A |
| 03 | INX | B |
| 13 | INX | D |
| 23 | INX | H |
| 33 | INX | SP |

### DECREMENT**

| | | |
|---|---|---|
| 05 | DCR | B |
| 0D | DCR | C |
| 15 | DCR | D |
| 1D | DCR | E |
| 25 | DCR | H |
| 2D | DCR | L |
| 35 | DCR | M |
| 3D | DCR | A |
| 0B | DCX | B |
| 1B | DCX | D |
| 2B | DCX | H |
| 3B | DCX | SP |

### MOVE

| | | |
|---|---|---|
| 40 | MOV | B,B |
| 41 | MOV | B,C |
| 42 | MOV | B,D |
| 43 | MOV | B,E |
| 44 | MOV | B,H |
| 45 | MOV | B,L |
| 46 | MOV | B,M |
| 47 | MOV | B,A |
| 48 | MOV | C,B |
| 49 | MOV | C,C |
| 4A | MOV | C,D |
| 4B | MOV | C,E |
| 4C | MOV | C,H |
| 4D | MOV | C,L |
| 4E | MOV | C,M |
| 4F | MOV | C,A |
| 50 | MOV | D,B |
| 51 | MOV | D,C |
| 52 | MOV | D,D |
| 53 | MOV | D,E |
| 54 | MOV | D,H |
| 55 | MOV | D,L |
| 56 | MOV | D,M |
| 57 | MOV | D,A |

### PSEUDO INSTRUCTION

| | | |
|---|---|---|
| ORG | Adr | |
| END | | |
| EQU | D16 | |
| SET | D16 | |
| DS | D16 | |
| DB | D8 | [ ] |
| DW | D16 | [ ] |
| IF | D16 | |
| ENDIF | | |
| MACRO | [ ] | |
| ENDM | | |

### CONSTANT DEFINITION

| | |
|---|---|
| 0BDH, 1AH | Hex |
| 105D, 105 | Decimal |
| 72O, 72Q | Octal |
| 11011B, 00110B | Binary |
| 'TEST', 'A' 'B' | ASCII |

### OPERATORS

( , )
* , / , MOD,SHL,SHR
+,−
NOT
AND
OR,XOR

### STANDARD SETS

| | | |
|---|---|---|
| A | SET | 7 |
| B | SET | 0 |
| C | SET | 1 |
| D | SET | 2 |
| E | SET | 3 |
| H | SET | 4 |
| L | SET | 5 |
| M | SET | 6 |
| SP | SET | 6 |
| PSW | SET | 6 |

D8 = constant, or logical/arithmetic expression that evaluates to an 8 bit data quantity.

* = all Flags (C,Z,S,P) affected

D16 = constant, or logical/arithmetic expression that evaluates to a 16 bit data quantity.

† = only CARRY affected

Adr = 16 bit address

** = all Flags except CARRY affected; (exception: INX & DCX affect no Flags)

| | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | NOP | | 28 | --- | | 50 | MOV | D,B | 78 | MOV | A,B | A0 | ANA | B | C8 | RZ | | F0 | RP | | | | |
| 01 | LXI | B,D16 | 29 | DAD | H | 51 | MOV | D,C | 79 | MOV | A,C | A1 | ANA | C | C9 | RET | | F1 | POP | PSW | | | |
| 02 | STAX | B | 2A | LHLD | Adr | 52 | MOV | D,D | 7A | MOV | A,D | A2 | ANA | D | CA | JZ | | F2 | JP | Adr | | | |
| 03 | INX | B | 2B | DCX | H | 53 | MOV | D,E | 7B | MOV | A,E | A3 | ANA | E | CB | --- | | F3 | DI | | | | |
| 04 | INR | B | 2C | INR | L | 54 | MOV | D,H | 7C | MOV | A,H | A4 | ANA | H | CC | CZ | Adr | F4 | CP | Adr | | | |
| 05 | DCR | B | 2D | DCR | L | 55 | MOV | D,L | 7D | MOV | A,L | A5 | ANA | L | CD | CALL | Adr | F5 | PUSH | PSW | | | |
| 06 | MVI | B,D8 | 2E | MVI | L,D8 | 56 | MOV | D,M | 7E | MOV | A,M | A6 | ANA | M | CE | ACI | D8 | F6 | ORI | D8 | | | |
| 07 | RLC | | 2F | CMA | | 57 | MOV | D,A | 7F | MOV | A,A | A7 | ANA | A | CF | RST | 1 | F7 | RST | 6 | | | |
| 08 | --- | | 30 | --- | | 58 | MOV | E,B | 80 | ADD | B | A8 | XRA | B | D0 | RNC | | F8 | RM | | | | |
| 09 | DAD | B | 31 | LXI | SP,D16 | 59 | MOV | E,C | 81 | ADD | C | A9 | XRA | C | D1 | POP | D | F9 | SPHL | | | | |
| 0A | LDAX | B | 32 | STA | Adr | 5A | MOV | E,D | 82 | ADD | D | AA | XRA | D | D2 | JNC | Adr | FA | JM | Adr | | | |
| 0B | DCX | B | 33 | INX | SP | 5B | MOV | E,E | 83 | ADD | E | AB | XRA | E | D3 | OUT | D8 | FB | EI | | | | |
| 0C | INR | C | 34 | INR | M | 5C | MOV | E,H | 84 | ADD | H | AC | XRA | H | D4 | CNC | Adr | FC | CM | Adr | | | |
| 0D | DCR | C | 35 | DCR | M | 5D | MOV | E,L | 85 | ADD | L | AD | XRA | L | D5 | PUSH | D | FD | --- | | | | |
| 0E | MVI | C,D8 | 36 | MVI | M,D8 | 5E | MOV | E,M | 86 | ADD | M | AE | XRA | M | D6 | SUI | D8 | FE | CPI | D8 | | | |
| 0F | RRC | | 37 | STC | | 5F | MOV | E,A | 87 | ADD | A | AF | XRA | A | D7 | RST | 2 | FF | RST | 7 | | | |
| 10 | --- | | 38 | --- | | 60 | MOV | H,B | 88 | ADC | B | B0 | ORA | B | D8 | RC | | | | | | | |
| 11 | LXI | D,D16 | 39 | DAD | SP | 61 | MOV | H,C | 89 | ADC | C | B1 | ORA | C | D9 | --- | | | | | | | |
| 12 | STAX | D | 3A | LDA | Adr | 62 | MOV | H,D | 8A | ADC | D | B2 | ORA | D | DA | JC | Adr | | | | | | |
| 13 | INX | D | 3B | DCX | SP | 63 | MOV | H,E | 8B | ADC | E | B3 | ORA | E | DB | IN | D8 | | | | | | |
| 14 | INR | D | 3C | INR | A | 64 | MOV | H,H | 8C | ADC | H | B4 | ORA | H | DC | CC | Adr | | | | | | |
| 15 | DCR | D | 3D | DCR | A | 65 | MOV | H,L | 8D | ADC | L | B5 | ORA | L | DD | --- | | | | | | | |
| 16 | MVI | D,D8 | 3E | MVI | A,D8 | 66 | MOV | H,M | 8E | ADC | M | B6 | ORA | M | DE | SBI | D8 | | | | | | |
| 17 | RAL | | 3F | CMC | | 67 | MOV | H,A | 8F | ADC | A | B7 | ORA | A | DF | RST | 3 | | | | | | |
| 18 | --- | | 40 | MOV | B,B | 68 | MOV | L,B | 90 | SUB | B | B8 | CMP | B | E0 | RPO | | | | | | | |
| 19 | DAD | D | 41 | MOV | B,C | 69 | MOV | L,C | 91 | SUB | C | B9 | CMP | C | E1 | POP | H | | | | | | |
| 1A | LDAX | D | 42 | MOV | B,D | 6A | MOV | L,D | 92 | SUB | D | BA | CMP | D | E2 | JPO | Adr | | | | | | |
| 1B | DCX | D | 43 | MOV | B,E | 6B | MOV | L,E | 93 | SUB | E | BB | CMP | E | E3 | XTHL | | | | | | | |
| 1C | INR | E | 44 | MOV | B,H | 6C | MOV | L,H | 94 | SUB | H | BC | CMP | H | E4 | CPO | Adr | | | | | | |
| 1D | DCR | E | 45 | MOV | B,L | 6D | MOV | L,L | 95 | SUB | L | BD | CMP | L | E5 | PUSH | H | | | | | | |
| 1E | MVI | E,D8 | 46 | MOV | B,M | 6E | MOV | L,M | 96 | SUB | M | BE | CMP | M | E6 | ANI | D8 | | | | | | |
| 1F | RAR | | 47 | MOV | B,A | 6F | MOV | L,A | 97 | SUB | A | BF | CMP | A | E7 | RST | 4 | | | | | | |
| 20 | --- | | 48 | MOV | C,B | 70 | MOV | M,B | 98 | SBB | B | C0 | RNZ | | E8 | RPE | | | | | | | |
| 21 | LXI | H,D16 | 49 | MOV | C,C | 71 | MOV | M,C | 99 | SBB | C | C1 | POP | B | E9 | PCHL | | | | | | | |
| 22 | SHLD | Adr | 4A | MOV | C,D | 72 | MOV | M,D | 9A | SBB | D | C2 | JNZ | Adr | EA | JPE | Adr | | | | | | |
| 23 | INX | H | 4B | MOV | C,E | 73 | MOV | M,E | 9B | SBB | E | C3 | JMP | Adr | EB | XCHG | | | | | | | |
| 24 | INR | H | 4C | MOV | C,H | 74 | MOV | M,H | 9C | SBB | H | C4 | CNZ | Adr | EC | CPE | Adr | | | | | | |
| 25 | DCR | H | 4D | MOV | C,L | 75 | MOV | M,L | 9D | SBB | L | C5 | PUSH | B | ED | --- | | | | | | | |
| 26 | MVI | H,D8 | 4E | MOV | C,M | 76 | HLT | | 9E | SBB | M | C6 | ADI | D8 | EE | XRI | D8 | | | | | | |
| 27 | DAA | | 4F | MOV | C,A | 77 | MOV | M,A | 9F | SBB | A | C7 | RST | 0 | EF | RST | 5 | | | | | | |

D8 = constant, or logical/arithmetic expression that evaluates to an 8 bit data quantity.

D16 = constant, or logical/arithmetic expression that evaluates to a 16 bit data quantity.

Adr = 16 bit address

**HEX-ASCII TABLE**

| Printing | | Characters | |
|---|---|---|---|
| 30 | 0 | 40 | @ |
| 31 | 1 | 20 | space |
| 32 | 2 | 21 | ! |
| 33 | 3 | 22 | " |
| 34 | 4 | 23 | # |
| 35 | 5 | 24 | $ |
| 36 | 6 | 25 | % |
| 37 | 7 | 26 | & |
| 38 | 8 | 27 | ' |
| 39 | 9 | 28 | ( |
| | | 29 | ) |
| 41 | A | 2A | * |
| 42 | B | 2B | + |
| 43 | C | 2C | , |
| 44 | D | 2D | − |
| 45 | E | 2E | . |
| 46 | F | 2F | / |
| 47 | G | 3A | : |
| 48 | H | 3B | ; |
| 49 | I | 3C | < |
| 4A | J | 3D | = |
| 4B | K | 3E | > |
| 4C | L | 3F | ? |
| 4D | M | 5B | [ |
| 4E | N | 5C | \ |
| 4F | O | 5D | ] |
| 50 | P | 5E | ↑ (∧) |
| 51 | Q | 5F | ← (−) |
| 52 | R | | |
| 53 | S | | |
| 54 | T | | |
| 55 | U | | |
| 56 | V | | |
| 57 | W | | |
| 58 | X | | |
| 59 | Y | | |
| 5A | Z | | |

**HEX-ASCII TABLE**

Non-Printing

| | |
|---|---|
| 00 | NULL |
| 07 | BELL |
| 09 | TAB |
| 0A | LF |
| 0B | VT |
| 0C | FORM |
| 0D | CR |
| 11 | X-ON |
| 12 | TAPE |
| 13 | X-OFF |
| 14 | |
| 1B | ESC |
| 7D | ALT MODE |
| 7F | RUB OUT |

## DIM-1001 COMPUTER MODULE

The DIM-1001 Computer module is a complete "computer on a board"; containing:

- "One chip eight bit parallel CPU" (Intel 8080), with 74 basic processor instructions. 2$\mu$ sec instruction cycle.
- 2kByte of Programmable/Erasable read only memory (PROM).
- 2kByte of mask programmable read only memory (ROM).
- 1kByte of random access memory (RAM).
- Address Bus Drivers providing 65kBytes direct addressing capability.
- Interrupt control. Eight priority levels with program controlled enable/disable.
- Teletype or displayterminal interface. Selectable Baud Rate 110, 1200, 2400, 4800, 9600.
- Input/Output port control signals providing 256 Input and 256 Output port capability.
- Circuits providing DMA capability.

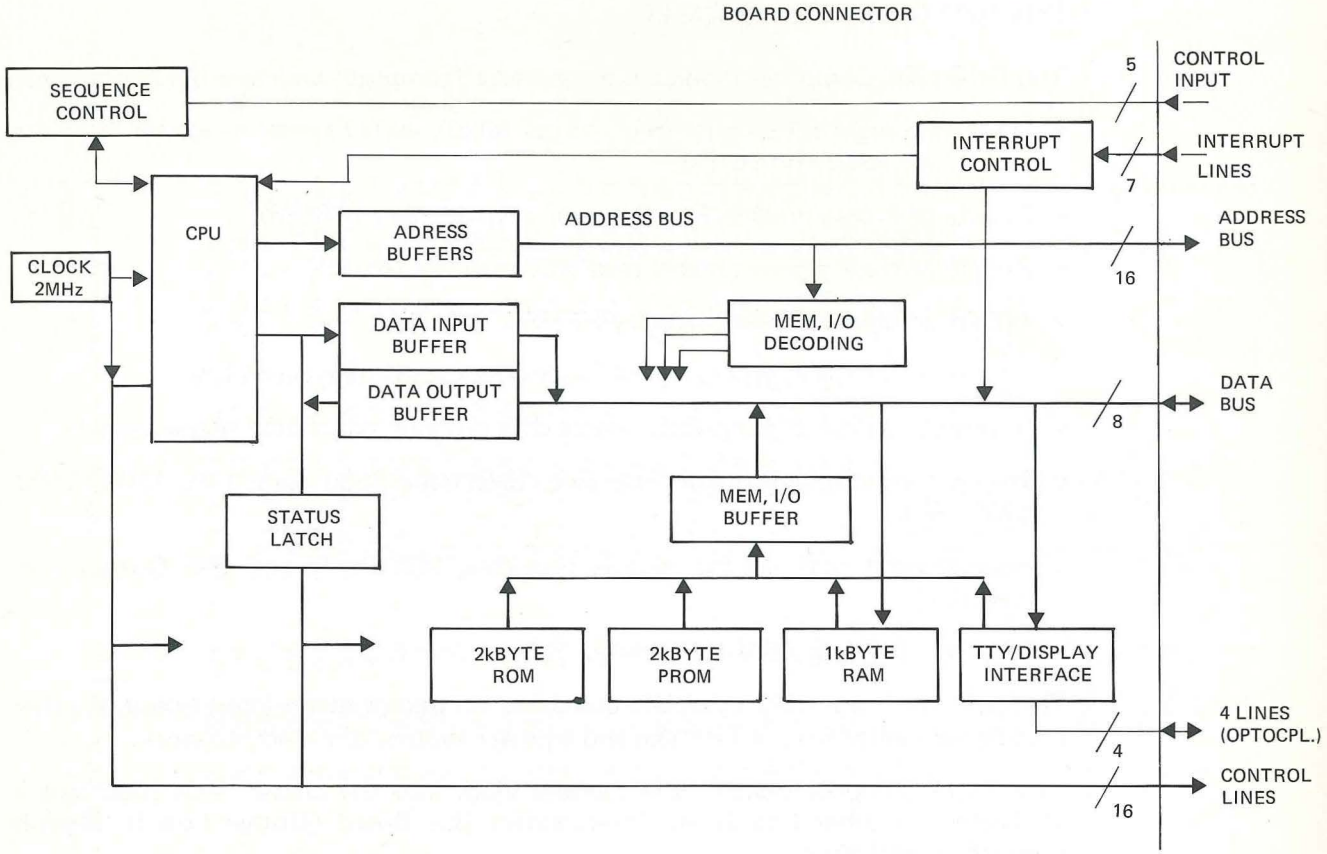
The DIM-1001 is a truly complete computer on one printed circuit board. By connecting four wires from a Teletype and a power supply, it is ready to work.

In a more complex system, the module plugs into the chassis DIM-1090 and is connected to other boards via Interconnect Bus Board (Mother board). Module position is irrelevant.

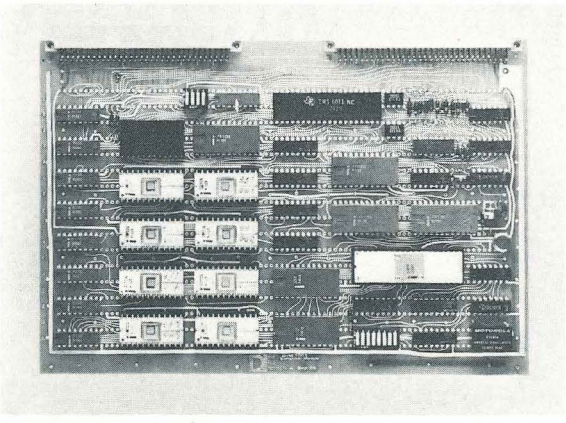
The DIM-1001 has a 16-bit TTL compatible memory address bus, an 8-bit bidirectional data bus and a control bus providing memory read and write signals, input/output control signals, interrupt control and DMA control signals.

A crystal controlled system clock running at 2 MHz is included on the board.

Asynchroneous interface between CPU and memory is ensured by a READY signal. This allows any combination of memory speeds and any type of memory in any combination to be interfaced at TTL level (ROM, PROM, RAM).




DIM-1001 Simplified Block Diagram



## DIM-1010 4kByte RAM MEMORY MODULE

The DIM-1010 RAM Memory Module is a random access memory module containing 4096 8-bit bytes. The module has address and data buffers, read/write timing circuits and full 16-bit address decoding.

The Module has address and data buffers, read/write timing circuits and full 16-bit address decoding.

The specific location in the 65 kByte memory space is selected in 4 kByte steps by setting bits 12-15 to the proper value. A miniature switch in an 8-pin dual-in-line package is included for this purpose.

Cycle time is 700 nanoseconds. Power consumption is 1 A at 5V. No other supplies are required. The memory is static.

## DIM-1012 4kByte PROM MEMORY MODULE

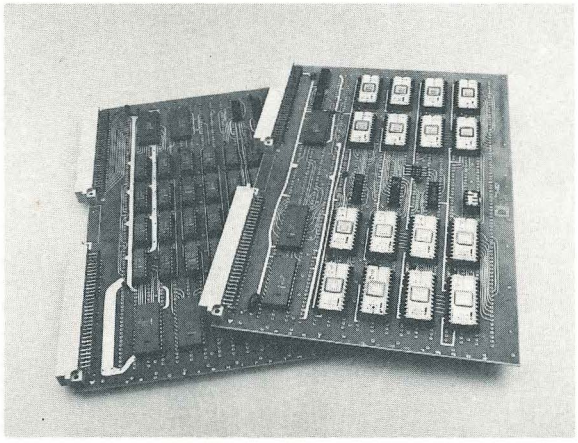
The DIM-1012 PROM Memory Module is a programmable read only memory module containing 4096 8-bit bytes. It has sockets for 16 Intel 1702A or 8702. These circuits are reprogrammable. In addition the mask programmed 1302 og 8302 can be used in place of the PROMs when so desired.

The specific location in the 65 kByte memory space is selected in 4 kBytes steps by setting bits 12-15 to the proper value. For this purpose a miniature switch in a dual-in-line package is included on the board.

The module has address and data buffers and the necessary timing circuits.

Power consumption is 200 mA from 5V and 600 mA from -9V.

DIM-1010
4kByte RAM
Memory Module



DIM-1012
4kByte PROM
Memory Module

## DIM-1013 16 kByte RAM MEMORY MODULE

The DIM-1013 RAM Memory Module is a random access memory module containing 16384 8-bit bytes, based on 4 k bit dynamic memories.

The module has address and data buffers, read/write timing circuits and full 16-bit address decoding. Refresh circuits are included on the board. The CPU will not be affected by the refresh, there will only be an increase in access time when a CPU request coinsides with a refresh.

The specific location in the 65 k memory space is selected by simple straps on Address bits 14-15.

Power consumption is 0.5A from 5V, 0.2A from 12V and 5mA from -5V.

## DIM-1030 DISC CONTROLLER (4/256 kByte)

The DIM-1030 Disc Controller is a module providing the control and interface for up to four floppy disc drives. The controller assures compatibility with the IBM 3740 Data Entry System and the 3540 Diskette Input/Output Unit.

The device is realized on one single circuit board by making extensive use of the computer module DIM-1001. Setting of control registers, reading of status registers as well as the 250 kbaud data transfer between the disc drive and the system is performed by input/output instructions. Thus, the amount of hardware is reduced and a flexibility is achieved that enables the controller to be interfaced at various disc drives by changing programs.

Drivers and line receivers are provided for, based on a maximum distance of 20 feet.

A listing of a disc drive control program is delivered as a part of the customer documentation for the DIM-1030 Disc Controller.

Complete software support is available.

## DIM-1021 QUAD PERIPHERAL I/O MODULE

The DIM-1021 is a module providing 3 latching input ports and 3 latching output ports. Device numbers are individually selctable by switches on the board.

Line drivers sinking 100mA and receivers with input signal range ± 30V are provided.

Each port has input and output flag to external device. Polarity of flags are switch selectable. Status Register indicating Data Ready/Transmission Ready is incorporated.

Typical applications for the DIM-1021 are interface for Paper Tape Reader, Paper Tape Punch, Magnetic Tape Cassettes, etc.

## DIM-1090 CHASSIS WITH MOTHERBOARD

The DIM-1090 Chassis is designed as a convenient compartment for a medium sized MYCRO-1. It consists of a Bottom Support, a Circuit Board Block with a Mother-board for interconnection and a Cover. The DIM-1090 Chassis is delivered with or without the DIM-1091 Power Supply.

It can be a desk-top unit, rack-mountable or it could simply be put inside the base of your teletype.

The Circuit Board Block can take 6 Double Europe-Standard boards. As an example this enables you to have a CPU and 35 kByte of memory and two spare slots.

## DIM-1091 POWER SUPPLY WITH SWITCH PANEL

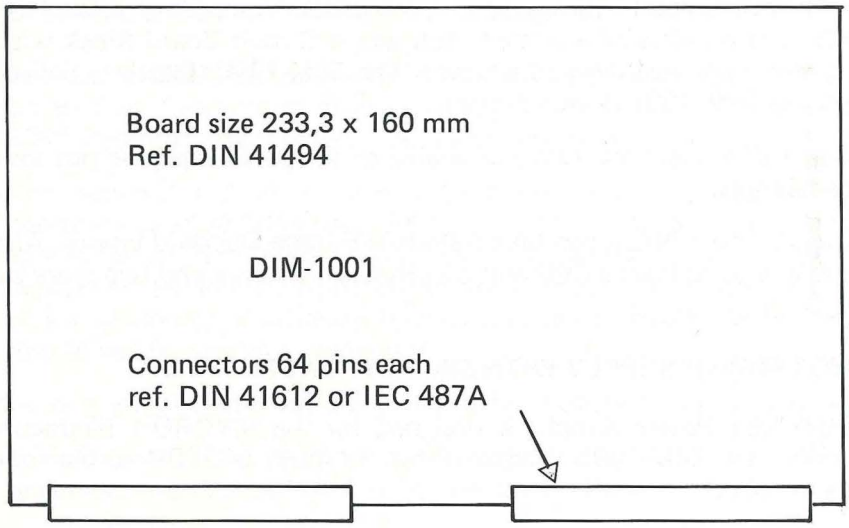
The DIM-1091 Power Supply is designed for the MYCRO-1 Microcomputer and mounted in the DIM-1090 Chassis. From an input of 220V ac the following voltages are supplied:

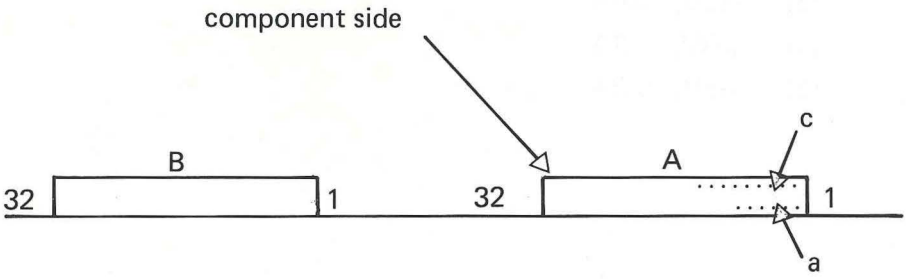
(1) 5V, 8A
(2) 12V, 0.5A
(3) −12V, 0.5A
(4) −9V, 2A
(5) −5V, 0.2A

## DIM-1001 MOTHERBOARD INTERFACE

The table below contains a definition of the input and output signals on the board connectors.

All levels are TTL-compatible unless otherwise noted. Positive logic is used.



## DIM-1001 CONNECTOR DEFINITION

| PIN NO. | Notation | Description |
|---|---|---|
| A1 a,c | – 12V | –12V Supply. Load 50 mA. (30 mA when TTY current loop is not active.) Normally strapped to pin B13a. (May be cut if separate supplies are desired.) |
| A2a,c | –9V | –9V Supply. Load 200 mA. |
| A3,c | $\overline{\text{WEM}}$ | Active low output signal enabling a write memory operations. Sinks 6 mA. |
| A4c<br>–<br>A14c | AB0<br>–<br>AB10 | Address bus lines. Sink capability 14 mA. High Impedance (THREE-STATE) when HOLD; ref. $\overline{\text{HLDA}}$. |
| A15c | STACK | Status output signal indicating that the Address Bus holds a stack address from the STACK POINTER register. Sinks 15 mA. High Impedance when HOLD, ref. $\overline{\text{HLDA}}$. |
| A16c<br>A17c<br>A18c<br>A19c<br>A20c<br>A21c<br>A22c<br>A23c | $\overline{\text{INT 3}}$<br>$\overline{\text{INT 2}}$<br>$\overline{\text{INT 1}}$<br>$\overline{\text{INT 0}}$<br>$\overline{\text{INT 7}}$<br>$\overline{\text{INT 6}}$<br>$\overline{\text{INT 5}}$<br>$\overline{\text{INT 4}}$ | Interrupt Input lines. Active low, load 1,6 mA each. Int X gives restart address x 8. The lower have priority. |
| A25c<br>–<br>A29c | AB11<br>–<br>AB15 | Address bus lines. Ref. A4c – A14c. |
| A30c | HLTA | Status output signal indicating that the CPU is in a HALT (programmed stop) state. Sinks 15 mA High Impedance when HOLD, ref. $\overline{\text{HLDA}}$. |
| A31c | INP | Status output signal indicating that the Address Bus bits 8-15 hold an Input Device number. Data are transferred to Accumulator from the Data Bus when $\overline{\text{DBIN}}$ (ref. B22c) goes from low to high. Sinks 12 mA. |
| A32c | M1 | Status output Signal indicating that the CPU is in the Instruction fetch state. Sinks 15 mA. High Impedance when HOLD, ref. $\overline{\text{HLDA}}$. |
| B1c<br>–<br>B8c | DB7<br>–<br>DB0 | Bidirectional Data Bus bits 7-0, controlled by $\overline{\text{DBIN}}$ (ref. B22c). Sink capability is 12 mA, Input load 2 mA. High Impedance when HOLD, ref. HLDA. |

## DIM-1001 CONNECTOR DEFINITION (Cont'd)

| PIN NO. | Notation | Description |
|---|---|---|
| B9c | CLOCK | TTL level output. 2 MHz clock in phase with Ø 1 CPU clock. Frequency stability. |
| B10c | MEMR | Status output signal indicating that current cycle is a memory read operation. Sink capability is 12 mV. High Impedance when HOLD, ref. $\overline{\text{HLDA}}$. |
| B11c | $\overline{\text{HLDA}}$ | Hold Acknowledge Output Inverted. Sinks 8 mA. |
| B12c | INTA | Interrupt Acknowledge status output. Sinks 15 mA. High Impedance when HOLD ref. $\overline{\text{HLDA}}$. |
| B13a | −12V | Optional −12V Power Supply for Teletype/ Display Interface. This interface is optocoupled. If a remote ± 12V supply for the interface is desirable, the connection to A1a,c may be broken. |
| B13c | $\overline{\text{WO}}$ | Active low status output signal indicating that current cycle is write memory or a write output operation. Sinks 15 mA. High Impedance when HOLD, ref. $\overline{\text{HLDA}}$. |
| B14a<br>B15a | −TTY KEY<br>+TTY KEY | 20 mA Current Loop Input from Keyboard. |
| B14c | OUT | Status output signal indicating that the Address Bus bits 8-15 holds an output device number. Accumulator data are transferred when $\overline{\text{WE}}$ is low. Sinks 12 mA. High Impedance when HOLD, ref. $\overline{\text{HLDA}}$. |
| B15c | $\overline{\text{WE}}$ | Active low output signal enabling a write operation. This signal is also active in a write memory operation. Sinks 6 mA. |
| B16a | +12V | Optional +12V Supply. Connection to B30 a, c may be broken if separate supply for TTY/Display interface is desired. Ref −12V, pins A1a, c and B13a. |
| B16c | $\overline{\text{WEX}}$ | Auxiliary memory write enable signal for use in DMA. |
| B17a<br>B18a | −TTY PRINT<br>+TTY PRINT | 20 mA current loop output to Teletype printer. |
| B17c | $\overline{\text{OUTTY}}$ | Decoded enable signal for outptu TTY/DISPLAY |
| B18c | $\overline{\text{DBE}}$ | Auxiliary data to Data Bus enable signal for use in DMA. |
| B19c | HOLD | HOLD request input. Load 0,01 mA. |

## DIM-1001 CONNECTOR DEFINITION (Cont'd)

| PIN NO. | Notation | Description |
|---|---|---|
| B20c | INT | INTERRUPT REQUEST. Test output. Sinks 8 mA. |
| B21c | INTE | Output signal indicating that Interrupt is enabled. (Interrupt flip-flop is set.) |
| B22c | $\overline{\text{DBIN}}$ | Active low control output signal. Data bus content are transferred to CPU when $\overline{\text{DBIN}}$ is active. Sinks 7 mA. |
| B23c | $\overline{\text{READY}}$ | Active low input signal used for identification of memory and In/Out devices to CPU. Load 1.6 mA. |
| B24c | $\overline{\text{XRESET}}$ | Active Low Input Signal, when Active the Program counter and Instruction register is set to zero. After $\overline{\text{XRESET}}$ the CPU will start a fetch cycle in location 0 in memory. Load 1.6 mA. |
| B25c | $\overline{\text{STEP}}$<br>$\overline{\text{DELAY}}$ | Input control signals for use in memory and In/Out device synchronization and in single cycle mode. $\overline{\text{DELAY}}$ active (low) disables normal sequencing. When $\overline{\text{DELAY}}$ is active (low) a pulse on the $\overline{\text{STEP}}$ input causes the CPU to proceed one cycle. Triggering is on the positive edge of the pulse. Load 1.6 mA. |
| B27c | $\overline{\text{SYNC}}$ | Active low output indicating the beginning of a CPU cycle. Sinks 6 mA. |
| B28c | $\overline{\text{WAIT}}$ | Active low output signal indicating that the CPU is in a wait state. Sinks 8 mA. |
| B29a,c | −5V | −5Volts supply, load 1 mA. |
| B30a,c | +12V | +12 Volts supply, load 60 mA (40 mA when TTY Current loop is inactive). |
| B31a,c | GND | Power and Signal GND |
| B32a,c | +5 | +5 Volts supply, load 1.2 A. |

## DIM-1001 CONNECTOR DEFINITION (SUMMARY)

| A | | | B | | |
|---|---|---|---|---|---|
| a | | c | a | | c |
| – 12V | 1 | – 12V | | 1 | DB7 |
| – 9V | 2 | – 9V | | 1 | DB6 |
| | 3 | $\overline{\text{WEM}}$ | | 3 | DB5 |
| | 4 | | | 4 | DB4 |
| | | | | | DB3 |
| | 5 | AB1 | | 5 | DB3 |
| | 6 | AB2 | | 6 | DB2 |
| | 7 | AB3 | | 7 | DB1 |
| | 8 | AB4 | | 8 | DB0 |
| | 9 | AB5 | | 9 | CLK (Ø1) |
| | 10 | AB6 | | 10 | $\overline{\text{MEMR}}$ |
| | 11 | AB7 | | 11 | $\overline{\text{HLDA}}$ |
| | 12 | AB8 | | 12 | INTA |
| | 13 | AB9 | (– 12V) (1) | 13 | $\overline{\text{WO}}$ |
| | 14 | AB10 | –TTY KEY | 14 | $\overline{\text{OUT}}$ |
| | 15 | STACK | +TTY » | 15 | $\overline{\text{WE}}$ |
| | 16 | $\overline{\text{INT 3}}$ | (+ 12V) (2) | 16 | $\overline{\text{WEX}}$ |
| | 17 | $\overline{\text{INT 2}}$ | –TTY PRINT | 17 | $\overline{\text{OUTTY}}$ |
| | 18 | $\overline{\text{INT 1}}$ | +TTY » | 18 | $\overline{\text{DBE}}$ |
| | 19 | $\overline{\text{INT 0}}$ | | 19 | HOLD |
| | 20 | $\overline{\text{INT 7}}$ | | 20 | INT |
| | 21 | $\overline{\text{INT 6}}$ | | 21 | INTE |
| | 22 | $\overline{\text{INT 5}}$ | | 22 | $\overline{\text{DBIN}}$ |
| | 23 | $\overline{\text{INT 4}}$ | | 23 | $\overline{\text{READY}}$ |
| | 24 | | | 24 | XRESET |
| | 25 | AB11 | | 25 | $\overline{\text{STEP}}$ |
| | 26 | AB12 | | 26 | $\overline{\text{DELAY}}$ |
| | 27 | AB13 | | 27 | $\overline{\text{SYNC}}$ |
| | 28 | AB14 | | 28 | $\overline{\text{WAIT}}$ |
| | 29 | AB15 | – 5V | 29 | – 5V |
| | 30 | HLTA | + 12V | 30 | + 12V |
| | 31 | INP | GND | 31 | GND |
| | 32 | M1 | + 5V | 32 | + 5V |

# Released just before press time!

## MYCRO-1 FLOATING POINT ARITHMETIC AND I/O PACKAGE

The floating point package developed for MYCRO-1 has a format which allows a variable length mantissa and has one byte exponent. The user can choose the number of bytes in the mantissa to match his need for numeric precision. The mantissa has a packed decimal format with two figures in each byte. Bit 7 in the exponent is the sign bit, bit 6 gives the complement of the exponent's sign. (+ : = 1, – : = 0) The largest magnitude that can be represented is $10^{+126}$, $10^{-128}$.

The complete package occupies nearly 2k og PROM and less than 1/4k of RAM. The floating point package is independent of other software.

The system consists of the following subroutines:

- FAD – Floating point addition
- FSB – Floating point subtraction
- FMU – Floating point multiplication
- FDV – Floating point division

When FAD, FSB, FMU and FDV are called the BC-register must contain pointer to destination, DE-register pointer to source 1 and HL-register pointer to source 2.

- NLZ – A two bytes fixed point number normalized to a floating point number
- DNZ – A floating point number denormalized to a two bytes fixed point number. The carry flip-flop is set to one when overflow.

When NLZ and DNZ are called the BC-register must contain pointer to destination and DE-register pointer to source.

- FLTIN – Floating point input in free format. The carry flip-flop is set to 1 when correct syntax and 0 when error.

When FLTIN is called the BC-register must contain a pointer to data destination.

- FLTUT – Floating point output, with a given format.

When FLTUT is called the BC-register must contain a pointer to data and E-register the number of data to be output.

- FORM – Specify format for FLTUT. When called the C-register must contain the total field and E-register the fraction. The output format must be specified before FLTUT is called.

Universitetsbiblioteket i Oslo
Det matematisk-naturvitenskapelige fakultetsbibliotek
009408NA0